Learning a Contextual and Topological Representation of Areas-of-Interest for On-Demand Delivery Application

Mingxuan Yue ✉[1], Tianshu Sun[1], Fan wu[2], Lixia Wu[2], Yinghui Xu[2], and Cyrus Shahabi[1]

[1] University of Southern California, Los Angeles, U.S.
{mingxuay, shahabi}@usc.edu, tianshus@marshall.usc.edu
[2] Cainiao Network, Hangzhou, China
{wf118503, wallace.wulx}@cainiao.com

Abstract. A good representation of urban areas is of great importance in on-demand delivery services such as for ETA prediction. However, the existing representations learn either from sparse check-in histories or topological geometries, thus are either lacking coverage and violating the geographical law or ignoring contextual information from data. In this paper, we propose a novel representation learning framework for obtaining a unified representation of Area of Interest from both contextual data (trajectories) and topological data (graphs). The framework first encodes trajectories and graphs into homogeneous views, and then train a multi-view autoencoder to learn the representation of areas using a ranking-based loss. Experiments with real-world package delivery data on ETA prediction confirm the effectiveness of the model.

Keywords: Representation Learning; Trajectories; Multi-view autoencoder

1 Introduction

In recent years, we witness the rapid growth of on-demand deliveries everywhere and every day (e.g., Amazon Prime Now). We deliver people, food, parcels by cars, bicycles, and foot from dawn to midnight and from city centers to suburbans. The explosion of E-commerce and recent advances in spatial crowdsourcing have prompted the surge of deliveries, and are still calling for better solutions.

A good representation of spatial units is of vital importance to all delivery-related services [13]. Various companies like Uber and DiDi utilize different spatial extents such as grids, hexagons, or polygons to partition the space into spatial units [10]. These spatial units, represented by their coordinates and other geometric features, are then used as sources and targets for delivery services. Such spatial units fully cover an entire space (e.g., a city) and have nice topological properties. Thus, the algorithms based on these units can accommodate any possible delivery request. However, such topological representation can only capture spatial relationships between these units and ignore human's intuition and tacit knowledge on how to navigate between these regions. For

example, when couriers deliver packages on foot and/or by bike, they mainly choose paths according to their knowledge and experience on real-world road conditions and connections such as shortcuts, bridges, crowded streets, and crossings with long traffic lights. In such cases, mere topological representation often fails to capture key information thus may not be sufficient for the real delivery tasks. Fortunately, human trajectories capture such tacit knowledge and experiences.

Towards this end, recent work [3, 15, 32] strives to add such contextual data to Point Of Interest(POI) representation from check-in histories by adopting NLP models like Word2vec [17]. However, these studies mainly focus on recommending POI to users. Hence, the representation of POIs usually does not cover the entire space, thus they cannot be directly applied to delivery systems that requires every points in space to be reachable. Besides, the learned representation may also lose the topological property and conflict with the Tobler's First Law of Geography [26] which says "Everything is related to everything else, but near things are more related than distant things", due to the discrete locations of POIs and sampling bias in the collection of check-in histories.

Therefore, the best representation should learn from both topological and contextual data to take advantage of the best of the two worlds. To achieve this, we propose a novel Deep Multi-view informAtion-encoding RanKing-based network (DeepMARK) to learn a representation of spatial regions. Rather than regular-shaped regions, we consider the spatial regions to be geographically partitioned by map segmentation, i.e., the Areas of Interests (AOI) used in this paper. AOIs are non-overlapping irregular polygons that fully partition (and hence cover) the space and each AOI captures its individual context. For example, while hexagons or grids may split a school into two units or may have a unit containing multiple land uses, each AOI represents a single context.

To learn both the topological and contextual features of these AOIs, our proposed framework DeepMARK consists of three components: one to learn the topological representation, the second one to learn the contextual representation and finally the third to unify the first two components.

Contextual representation component: In the field of NLP, contextual representations are usually learned based on the distributional hypothesis [21] from real-world language sequences, i.e., human utterance. Analogous to NLP, for "spatial" context, we consider location sequences, i.e., human trajectories, as the data source from which we learn the contextual representation of AOIs. The trajectory data is selected for its relevance and scalability in learning contextual representations for delivery problems: 1) trajectories preserve human's knowledge and preferences in traveling between AOIs. 2) with the ubiquity of mobile devices and the prevalence of spatial crowdsourcing apps, trajectories can be easily collected at scale. To learn contextual representation from trajectories, we model the spatial distributional hypothesis using Pointwise Mutual Information (PMI) between AOIs calculated from trajectories. Subsequently, we learn a distributed representation based on the PMI using an autoencoder framework.

Topological representation component: To model topological properties of irregular-shaped AOIs, we define Euclidean graph and Adjacency graph to capture the spatial relationships of the AOIs. Lately, to learn representations from graphs, researchers proposed various graph embedding approaches [2, 7, 27, 20, 6]. Popular methods like Deepwalk [20] and Node2vec [6] are based on random walks and train the network on

randomly generated samples. However, in our problem, such a process cannot be easily trained with trajectories jointly. Therefore, we propose to estimate the node-wise mutual information in graphs and use the same autoencoder framework as used for trajectories to align the learning of the two heterogeneous views.

Unified representation component: Finally, to combine the two heterogeneous views, previous studies employ different strategies to model the correlation between views and control the learning across views [14, 24, 4]. However, none of these approaches could be directly applied to our problem because most of them are designed for text and image data. To the best of our knowledge, we are the first to study the joint learning of AOI representation using both trajectory and graph data. To join the learning of trajectories and graphs, we propose a novel multi-view autoencoder neural network that takes the PMI matrices generated by the previous two components and utilizes an innovative ranking-weighted loss to dynamically balance the learning between views.

We evaluated our representation with a large real-world package delivery data acquired from Cainiao Network. Our representation approach is shown to have up to 20% reduction of errors as compared to the adapted baseline approaches in predicting Estimated Time of Arrival (ETA) of real-world deliveries.

The remainder of the paper is organized as follows. Section 2 clarifies some basic definitions and important notations used in this paper. Section 3 presents the details of our proposed framework. In Section 4, we show the evaluation of our approach on real-world data. Finally, Section 5 introduces the related work followed by our conclusion in Section 6.

2 Preliminaries

In this section, we introduce some important concepts followed by the formal problem definition.

Definition 1 (Area Of Interest (AOI)). *An AOI is a minimum geographical unit in the form of a polygon. The raw AOIs are generated by partitioning a space with fine-grain road networks and geometric boundaries (e.g., roads, rivers, railways) using map segmentation techniques.*

By definition, the boundaries of AOIs, i.e., the irregular polygons can have different sizes and numbers of edges, which differentiate them from those of the conventional space partitioning techniques using regular shapes (e.g., hexagons). Moreover, our AOIs still do cover the entire space and each AOI captures a single context (e.g., a school). Later in Section 3.2 we show how we add latent features (learned from topological representations) to each AOI to enforce the Tobler's First Law of Geography.

Definition 2 (Trajectory). *A trajectory s is a sequence of spatio-temporal tuples $s = [s(1), s(2), \ldots, s(k), \ldots]$, where $s(k)$ is represented by a tuple consisting of the AOI v that contains the GPS point and a timestamp t, i.e., $s(k) = (v, t)$.*

We derive the modeling of contextual representation from the analogy in language models. Most word representation models explicitly or implicitly follow the distributional hypothesis introduced by linguists [21]. The hypothesis is often stated as: *words*

which are similar in meaning occur in similar contexts. In our problem, as sequences of AOIs (trajectories) are analogous to sequences of words (sentences), we make the following assumption:

Assumption 1 (Contextual representation of AOIs)**.** *A contextual representation of AOIs follows the spatial distributional hypothesis, that AOIs have similar contextual representations are usually visited closely and in a trip.*

Given the above definitions, we define our problem of learning a contextual and topological representation of AOIs as below.

Definition 3 (Learning a Contextual and Topological Representation of AOIs (CTRA) Problem). *Given a set of raw AOIs (i.e., without latent features)* V*, and a set of trajectories* S*, s.t.* $\forall s \in S, \forall (v,t) \in s, v \in V$*, the objective is to learn a mapping* $V \rightarrow Z$*, s.t., it generates a latent representation* $z \in Z$ *for each AOI* $v \in V$*, that follows the spatial distributional hypothesis, and Tobler's First Law of Geography.*

3 Methodology

We propose a Deep Multi-view informAtion-based RanKing network (DeepMARK) to solve the CTRA problem. DeepMARK consists of three parts: learning contextual representation, learning topological representation and jointly learning of both representations, which are elaborated in the following sections.

3.1 Learning contextual representation from trajectories

Modeling spatial distributional hypothesis The learning of contextual representation of AOIs in trajectories is analogous to the learning of word embeddings from sentences. To model the distributional hypothesis, word embedding techniques usually describe similarities between words using their contexts and then map words to hidden embeddings according to such similarities. For example, the word2vec model [17] maximizes the log probability as in Equation 1. The modeling of the context similarity is implicitly computed by predicting the context words (w_{t+j}) of a target word (w_t), which usually requires a sampling-based training process, e.g., negative sampling.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{1}$$

In this paper, rather than use the sampling-based training and objective, we propose to use Pointwise Mutual Information (PMI) to describe the contextual similarity between AOIs and learn the representation by decomposing the similarities using neural networks. We believe such approach has better compliance with CTRA problem because of the following reasons.

1. *The similarity is symmetric*. In word2vec models, people choose a center word and its context word to describe the similarity. In this case, the similarity of "A to B"

might be different that of "B to A", when choosing A or B as the center word. However, in delivery scenarios, we concern more about whether the 2 places are likely to be visited from each other. So we expect the similarity to be symmetric, i.e., $similarity(A, B) = similarity(B, A)$, which is guaranteed in PMI.

2. *The decomposition of PMI has comparable performance and is implicitly equivalent to SGNS.* As shown in recent studies, the SGNS model is implicitly factorizing the shifted PMI matrix [11] and a good decomposition of PMI(PPMI) matrix is comparable with word2vec models in various tasks. [12]
3. *The training process is easy for alignment in a multi-view learning framework.* Sampling-based training is hard to be extended to multi-view problems like CTRA. Even applying iterative training one cannot align different views well to the same training target (a single AOI) and train them jointly. However, the decomposition of PMI is easy for aligning the same AOI from different views which allows joint training described in Section 3.3

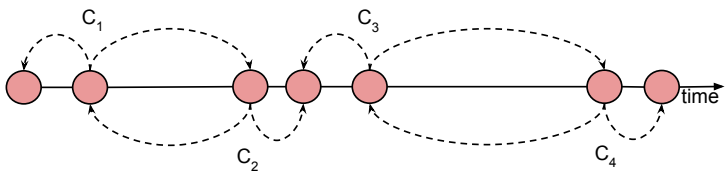
Formally, given AOI v_i and v_j, we define the contextual similarity from the trajectory data as follows:

$$\text{PMI}_{traj}(v_i, v_j) = log(\frac{p(v_i, v_j)}{p(v_i)p(v_j)}) \tag{2}$$

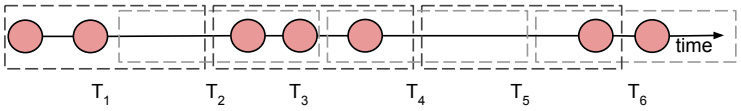
Here, $p(v_i)$ and $p(v_j)$ denote the probability of randomly visiting v_i and v_j, and $p(v_i, v_j)$ denotes the probability of visiting v_i and v_j together. we can interpret $\frac{p(v_i,v_j)}{p(v_i)p(v_j)}$ as: the ratio of *how likely people visit v_i and v_j together in the real world* to *how likely v_i and v_j are visited together at random*. Therefore, a large ratio means the two AOIs v_i and v_j are, rather than randomly visited together, co-visited for some real reason, e.g., they are easily accessible in human knowledge.

Computation of PMI in trajectories Now, to compute the PMI between AOIs, the remaining task is to define the computation of $p(v_i, v_j), p(v_i)$ and $p(v_j)$ for AOIs in trajectories. For calculating $p(v_i, v_j)$, it's important to properly define the co-occurrence of AOI v_i and AOI v_j in the trajectories. Different from the skip-gram model, we define that two AOIs co-occur in a close context if they fall in a fixed-length temporal window in a trajectory. To count such co-occurrences, we apply a time sliding window to each trajectory: $[t, t + \Delta]$, where Δ is the window size. As shown in Fig 1b, each sliding window may contain various numbers of AOIs (window T_1 has 2 AOIs while T_3 includes 3 AOIs) but the temporal length of each window is the same. In addition, we slide the windows with an offset of $\Delta/2$ to make the best use of the trajectories while avoid generating too many samples, similar to [31].

We use such temporal windows for defining co-occurrence because of the nature of trajectories. In detail, as depicted in Figure 1a, if we adopt the way skip-gram model building context windows (C_1 to C_4), for each AOI in the trajectory, we have to extract a fixed number of preceding and succeeding AOIs as its co-occurring neighbors. However, in trajectories, consecutively collected spatio-temporal points usually have variant time differences, e.g., from 2 min to 20 min, because of the unstable signals and different mobile application settings. Consequently, if we adopt skip-gram and consider two consecutive but distant AOIs as a co-occurrence, it will mislead the model to produce



(a) Count co-occurrences by skip-gram.



(b) Count co-occurrences by sliding window.

Fig. 1: Different ways of counting co-occurring AOIs

similar embeddings between the two distant AOIs (e.g., in C_3, two distant nodes are counted in the same context window), which is not expected.

After the sliding windows are generated, we count any two AOIs in the same window as a co-occurring pair. The probabilities $p(v_i, v_j)$, $p(v_i)$, $p(v_j)$, and the PMI matrix between AOIs can be estimated by counting the co-occurring pairs as below.

$$\begin{aligned}
\text{PMI}_{traj}(v_i, v_j) &= log(\frac{p(v_i, v_j)}{p(v_i)p(v_j)}) \\
&= log(\frac{\#(v_i, v_j)/|C|}{(\#(v_i)/|C|) \cdot (\#(v_j)/|C|)}) \\
&= log(\frac{\#(v_i, v_j) \cdot |C|}{\#(v_i) \cdot \#(v_j)}) \\
\text{where } |C| &= \sum_{i'} \sum_{j'} \#(v_{i'}, v_{j'})
\end{aligned}$$

In the equation above, $\#(v_i, v_j)$ denotes the count of co-occurring pairs (v_i, v_j) from all windows, $\#(v_i)$ and $\#(v_j)$ denotes the count of pairs containing v_i and v_j respectively. C denotes the set of all co-occurring pairs and $|C|$ is the number of all pairs.

Learning a distributed representation using autoencoder After the computation of PMI from trajectories, we propose to use autoencoder to decompose the PMI for a dense and distributed representation. Although for each AOI v_i, we can use its PMI similarity to all AOIs as its representation, i.e., $[\text{PMI}_{traj}(v_i, v_0), \text{PMI}_{traj}(v_i, v_1), \ldots \text{PMI}_{traj}(v_i, v_n)]$, we propose to apply low-rank decomposition by autoencoder on the sparse PMI matrix. Because a distributed representation [9](i.e., each element encodes multiple things) is always expressive and allows efficient activation in downstream training [1]. In addition, an autoencoder allows non-linear encoding, and thus could have more accurate

reconstruction of the similarities. Specifically, the autoencoder consists of an encoder f and a decoder g. The encoder f takes the PMI vector of each AOI and learns a low-dimensional embedding. Then the decoder g takes the low-dimensional embedding and reconstructs the PMI vector with minimum error. The objective of the network is minimizing the reconstruction error $\mathbb{L}$ in Equation 3.

$$\mathbb{L}_{traj} = \sum_{i}^{n} ||\text{PMI}_{traj}(i), g(f(\text{PMI}_{traj}(i)))||^2 \tag{3}$$

In summary, as depicted in Figure 2, DeepMARK first slides windows in trajectories, and then counts the co-occurring pairs in these windows. After that, the PMI matrix is computed based out of the counts, and is fed to an autoencoder for learning representations. Notice that here we actually employ the common practice of positive PMI [11, 12, 8] rather than PMI but we use PMI for simplicity.

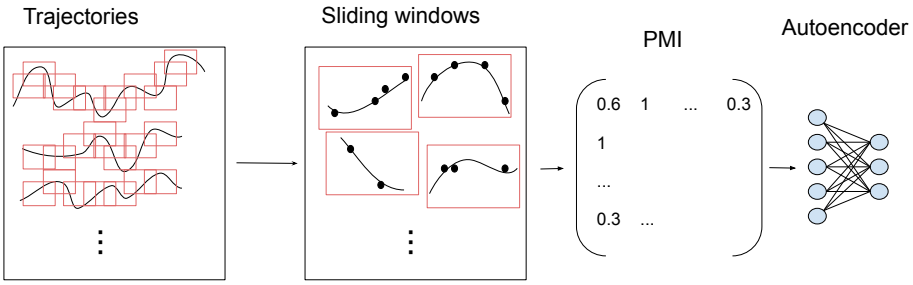


Fig. 2: Learn contextual representation from trajectories.

3.2 Learning topological representation from graphs

Given that our initial AOIs already cover the entire space (see Def. 2 in Section 2), here we would like to learn features (latent representation) per AOI to capture the spatial relationships among these AOIs that follow Tobler's first Law of Geography. Therefore, we use two graphs to capture the spatial relations between AOIs: Euclidean Graph G_{euc} and Adjacency Graph G_{adj}. Intuitively, the former graph captures Euclidean proximity to enforce Tobler's First Law between nearby AOIs and the latter uses adjacency relationships to enforce the law for adjacent AOIs.

Euclidean Graph G_{euc} We define $G_{euc} = \{V, E_{euc}, W_{euc}\}$. V is the set of nodes i.e., AOIs. We define the weights $W = [w_{ij}] \in \mathbb{R}^{n \times n}$ representing the proximity between v_i and v_j as a function of their Euclidean distance $dist(v_i, v_j)$. In particular, we define the proximity function as a thresholded Gaussian kernel function [22] as in Equation 4. Intuitively, the closer nodes, the larger weight is assigned to the edge between the nodes.

$$W_{ij} = \begin{cases} exp(-\frac{dist(v_i,v_j)^2}{\sigma^2}) & \text{if } dist(v_i, v_j) \leq \mathcal{K} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

Adjacency Graph G_{adj} We model the adjacency between AOIs as a graph $G_{adj} = \{V, E_{adj}, W_{adj}\}$. V is the set of nodes, i.e., AOIs. The weights $W = [w_{ij}] \in \{0, 1\}$ represent the adjacency between AOIs, where w_{ij} is defined as below.

$$W_{ij} = \begin{cases} 1 & \text{if } v_i \text{ is adjacent to } v_j \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

We expect the learning from the two graphs and from trajectories could have homogeneous processes for a flexible and alignable joint learning of topological and contextual representations. Therefore, we design PMI matrices for the graphs to be homogeneous with the trajectory view. For any two nodes v_i, v_j in a graph G given its weights W, to prepare the probabilities $p(v_i, v_j), p(v_i)$ and $p(v_j)$ in the graph, we define $p(v_i, v_j)$ as the proximity from v_i to v_j within K-step random walks. Specifically, we first define a transition matrix M^k, in which $M^k_{i,j}$ presents the probability of visiting v_j in a k step random walk from v_i with restart ratio η according to [25].

$$M^k = \eta \cdot \mathbb{I} + (1 - \eta) M^{(k-1)} \cdot (D^{-1} W),$$
$$\text{where } M^0 = \mathbb{I}$$

Here $\mathbb{I}$ is the identity matrix. D is a diagonal matrix, s.t., each element in the diagonal is the summation of the corresponding row in W, i.e., $D_{ii} = \sum_j W_{i,j}$. And respectively, as depicted in Figure 3, we can compute the proximity matrix P^K as the sum of random walks within K steps starting from any node: $P^K = \sum_{k=1}^{K} M^k$. Then $p(v_i, v_j)$ is defined as $P^K_{i,j}$, and accordingly, $p(v_i)$ is defined as $\sum_l P^K_{l,i}$.

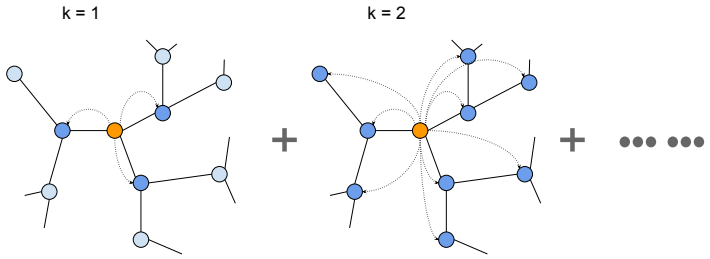


Fig. 3: Calculate proximity to other nodes by different-step random walks

Therefore we compute the PMI matrix for a graph $G = \{V, E, W\}$ with the maximum walking step k as below.

$$PMI^K_{graph}(v_i, v_j) = log(\frac{P^K_{i,j}}{\sum_l P^K_{l,i} \cdot \sum_l P^K_{l,j}})$$
$$\text{where } P^K = \sum_{k=1}^{K} M^k$$

After computing the PMI matrices for G_{euc} and G_{adj}, we can use the same autoencoder framework as for PMI$_{traj}$ to learn the topological representations from the graphs. And the remaining task is to jointly learn a representation from all autoencoders.

3.3 Jointly learning one representation by a multi-view ranking autoencoder

After the heterogeneous data are transformed into homogeneous views through different PMI computations, we propose to use a multi-view autoencoder for jointly learning the PMI in both trajectory and graphs as described in previous sections. In detail, all views (the PMI matrices) are fed into separate encoders and share the same middle layers, which generate embedding for the AOIs. Then separate decoders take the outputs of the shared layers and reconstruct the views and minimizing all errors. The network structure is depicted in Figure 4. For a straightforward multi-view autoencoder [18], the loss of our network could be written as a summation of all reconstruction losses:

$$\ell = (1 - \alpha - \beta) \cdot \mathbb{L}_{traj} + \alpha \cdot \mathbb{L}_{euc} + \beta \cdot \mathbb{L}_{adj} \tag{6}$$

Dynamic ranking-weighted loss In the equation 6, α, β are the weights of the two topological views (graphs). Rather than use static weights which require much effort in finding the optimal values and do not change during the training, we propose to dynamically change the weights according to the alignment between different views. Specifically, we expect the weight on the contextual view be correlated with the order-sensitive discordance between contextual and topological views. Below we provide the definition of such discordance:

Definition 4 (Order-sensitive Discordance). *For a given AOI v_i, an order-sensitive discordance between view A and view B happens if, the sorting of other AOIs by their similarities to v_i in view A is largely different from that in view B. In other words, AOI v_j ranks high in v_i's similarity sorted by view A, but ranks low in the sorting by view B.*

In this strategy, we introduce the inductive bias from real-world observations and domain knowledge. In detail, we observe that trajectories have different sampling density at different AOIs. Some AOIs and their neighbors are frequently visited in the trajectories. These AOIs have sufficient contextual semantics and are also consistent with the geography law. Then we want to learn more from (put more weight on) the contextual view. In contrast, some AOIs are rarely or never visited, and the learning from trajectories cannot learn a meaningful embedding from these AOIs. From the trajectory view, these AOIs cannot correctly order their relationships to other AOIs and would conflict the geographical law. In the latter case, we require more effort from the learning of graphs (higher weights on topological views) to ensure the law of geography.

Therefore, rather than use static values for α and β, we propose to use dynamic weights based on ListMLE [30], a list-wise ranking loss, computed between the graph PMI vectors and the trajectory PMI vector. If we denote x_{traj} as the reconstructed vector from trajectory view, y_{euc}, y_{adj} as the reconstructed vectors from G_{euc} and G_{adj}, $h_i(y_{euc})$ as the AOI index at the i_{th} largest value of y_{euc}, the ListMLE-based weights can be written in Equation 7. Intuitively, α and β are large if y_{euc}, y_{adj} have a different

ranking of elements from x_{traj}. In other words, the largest element in y_{euc} might be the smallest in x_{traj}. A possible example could be when both v_1 and v_2 are not visited in trajectories, the ranking of their similarity is based on a default value which could conflict with the ranking of the topological similarity learned from y_{euc} and y_{adj}. In this case, DeepMARK puts more weights on G_{euc} and G_{adj}.

$$\alpha = -\frac{1}{2}\sum_{j=1}^{n} log\frac{e^{x_{traj}(h_j(y_{euc}))}}{\sum_{k=j}^{n} e^{x_{traj}(h_k(y_{euc}))}} \tag{7}$$

$$\beta = -\frac{1}{2}\sum_{i=1}^{n} log\frac{e^{x_{traj}(h_j(y_{adj}))}}{\sum_{k=j}^{n} e^{x_{traj}(h_k(y_{adj}))}} \tag{8}$$

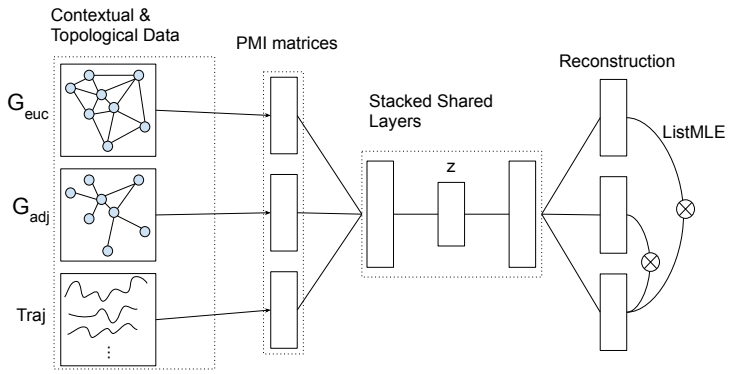


Fig. 4: DeepMARK Network Structure

4 Experiments

In this section, we evaluate the proposed model with real-world delivery datasets and task. We also include visualizations of interpretive results to help understand the model and the effect of different modules.

4.1 Dataset

We conduct the experiments on the package delivery data collected by Cainiao Network, handling more than a hundred million packages per day. In the experiment, the trajectory data is from a dispatching region from July 1, 2019 to Aug 31, 2019. The trajectories are pre-processed by removal of outliers, proper aggregation and mapped to AOIs. The original form of trajectories are GPS coordinates and timestamps, and after pre-processing, the input of this paper is sequences of AOIs and timestamps.

4.2 Experimental Settings

Adapted Baseline Algorithms Since there is no existing work on learning a contextual and topological AOI representation from trajectories and graphs, we adapted various approaches to our problem and compare them with DeepMARK. Here we list these adapted baseline approaches:

- Topological-only baselines:
  - **GeoHash** [19] is a general encoding of spatial objects. It maps the coordinates to fixed-length vectors in which common prefix usually infers close locations.
  - **Deepwalk** [20] and **Node2vec** [6] are state-of-the-art graph representation models which learn node embedding by skip-gram model from generated random walks.
- Contextual-only baseline:
  - **Word2vec** [17] is a word embedding approach that learns word representation from sentences. We adapt the model to our problem by treating trajectories as sentences and AOIs as words.
- Homogeneously integrated baselines:
  - We create two straightforward baselines **word2vec + deepwalk** and **word2vec + node2vec**, which are concatenations of word2vec embedding and graph embeddings. Thus these approaches also have the same input information as PTE (described below) and DeepMARK for a fair comparison.
- Heterogeneously integrated baseline:
  - **PTE** [24] is a heterogeneous embedding model that learns word embedding from both sentences and graphs based on a Heterogeneous Information Network Embedding (HINE) approach. We adapt this model to our problem by treating the trajectories of AOIs as the sentences of words and replacing their graphs with our graphs.

Parameter Settings For all random walk generations from graphs in Deepwalk, Node2vec and PTE, the walking length is set to 30, and walks per node is set to 30. Specifically, for Node2vec, p and q are set to 4 and 1. In DeepMARK, the revisiting ratio η for G_{euc} and G_{adj} is set to 0.1. The sliding window size is set to 20 min and the sliding offset is 10 min.

Training, Validation and Testing Following the principle of time-related prediction, we use the latter data for testing and the earlier for training and validation using 80-20 splits.

4.3 Evaluation with ETA Prediction

We evaluate our embedding framework in the prediction of the Estimated Time of Arrival(ETA) in the last-mile package delivery task. Predicting ETA in the last-mile deliveries is challenging because the couriers usually travel by non-motor vehicles and the environments are very complex. In this task, we use deepETA [29] as the prediction model and replace the spatial representation of AOIs (by default Geohash in deepETA) with embeddings from the listed approaches to evaluate their performances.

Evaluation metrics We utilize Rooted Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to evaluate the prediction performance on different embeddings. The smaller value indicates better performance in the prediction of ETA.

Comparison Results In Table 1 we compare the errors of ETA prediction using deepETA with different representations. We can observe that DeepMARK has a significant advance over all other baselines, inducing up to 20% reduction of errors. Its variant DeepMARK$_{static}$ which uses fixed weights on multiple views is worse than DeepMARK but slightly better than others. We also draw the curves of MAE of validation set versus the training epochs in deepETA using different representations. We can observe that the embedding by DeepMARK enables the model converge to the lowest validation error. And we can observe that PTE has a similar performance with word2vec+node2vec. Both have little control of coordinating different views, thus induce larger errors than DeepMARK.

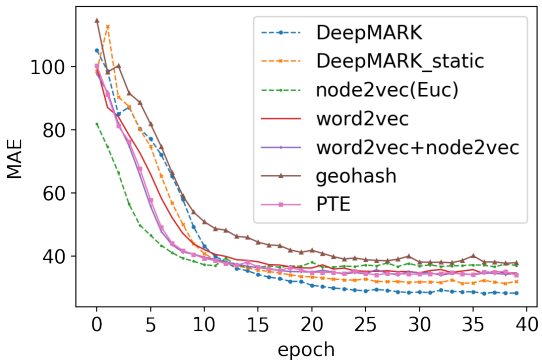


Fig. 5: Learning curves of ETA prediction by different representations

Table 1: ETA prediction performance

Model	RMSE (min)	MAE (min)
Geohash	55.22	40.98
word2vec	53.43	36.91
Deepwalk$_{euc}$	56.52	37.35
Deepwalk$_{adj}$	55.36	37.44
Node2vec$_{euc}$	54.76	37.12
Node2vec$_{adj}$	55.23	38.56
word2vec + deepwalk	54.15	37.41
word2vec + node2vec	53.86	36.53
PTE	53.17	36.52
DeepMARK$_{static}$	51.78	34.87
DeepMARK	**48.68**	**32.61**

4.4 Model Interpretation

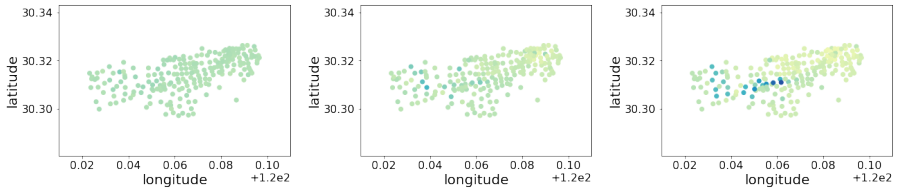
Visualization of the effect of joint learning We utilize t-SNE [16] to visualize the embeddings of AOIs by Word2vec on trajectories, Node2vec on G_{euc} and DeepMARK on both views in Figure 6. The colors of the points are based on Geohash values. That means points in similar colors are close in the real world. We can observe that in the Word2vec result, many distant AOIs are embedded closely (light yellow points and dark blue points). The Node2vec result has a smooth color transition from yellow to blue which indicates a nicely consistency with the law of geography, but the colors are almost evenly distributed which means it does not reveal any human knowledge on the AOIs. On the contrary, in DeepMARK result, the points have some variances in colors and shapes (holes and clusters in the figure) while overall the color transition is also smooth. This reflects that DeepMARK can learn human knowledge and meanwhile maintain the law of geography.

(a) Word2vec (b) Node2vec (c) DeepMARK

Fig. 6: Embedding visualization by t-SNE

Visualization of the changes of ranking-based weights To understand how the listMLE losses direct the training process of DeepMARK, we visualize the change of β (the ranking loss between the trajectory view and G_{adj} view) in Figure 7. In each plot, the x-axis and y-axis are the geometric coordinates, i.e., longitude and latitude. Each dot in the figures representing an AOI and its color denotes the value of β calculated for this AOI. dark blue indicates large β and shallow green indicates small β. In Figure 7 we show the calculated β of all AOIs at different training stages, i.e., epoch = 1, 30, 100. We can observe that: (a) The β for all AOIs have little difference at epoch 1 because of the randomness caused by initial parameters of the neural network; (b) The β of some AOIs get larger and others get smaller as the training proceeds to epoch 30; (c) A few AOIs have relatively large enough $beta$ while the majority of AOIs gain low $beta$ when the network is well-trained at epoch 100. Such change from epoch 0 to epoch 100 indicates the intuition behind our ranking-based weight strategy. Specifically, the ranking losses don't affect much in the reconstruction of all views at the early stages. Therefore it allows the model to have a warm start on roughly learning the representation of all views. However, when each view gets well trained and the topological views and contextual view become inconsistent in the ordering perspective, the ranking losses

(α and β) start to regularize these disordering according to our inductive bias until the reconstructions and the ranking losses across different views are balanced.



(a) Epoch 1 (b) Epoch 30 (c) Epoch 100

Fig. 7: Visualize the change of ranking loss between the topological view and the contextual view

5 Related Work

Point Of Interest Recommendation Lately, a few researchers spot their light on employing trajectory or (mostly) user check-in data in recommendation services, e.g., in [15, 3, 32]. These approaches utilize word2vec ideas for learning POI embeddings in the recommendation task. However, they are restricted to study embeddings for POIs which are discrete places thus not fully cover the space. And they didn't sufficiently consider the law of geography for the learned representation.

Graph representation learning As diverse real-world data could be formulated as graph structures, recent researchers study how to learn a graph representation to support different prediction and recommendation tasks. Inspired by word2vec models from studies in NLP, many researchers learn the node embedding in a graph by conducting random walks in the graph and treat the walks as corpus [20, 6]. In addition, some recent researches utilize deep neural networks to learn more neighboring information through an autoencoder framework [2, 27].

Multi-view representation learning Since real-world problems always involve different views of data, such as audio/video, image/text, multi-view representation learning attracts more attention in recent studies [14, 28]. In this area, researchers usually align different views by similarities or correlations or adopt different parameter sharing strategies to learn a representation [5, 18, 23]. Researchers also discover representation learning from heterogeneous graphs such as in [4]. Based on the heterogeneous graph embedding approaches, a recent study [24] proposed an approach that learns from sentences as well as graphs for text representation. However, the adaption of these approaches may not perform as good in delivery tasks due to the domain-specific alignment requirements in the spatial scenario.

6 Conclusion

In this paper, we introduced DeepMARK, an innovative deep multi-view autoencoder framework which learns a representation of AOI from trajectories and graphs data. The framework learns embedding of AOI that takes the best of both contextual and topological representations, i.e., incorporates data-driven contextual information and follows the Tobler's First Law of Geography. DeepMARK is evaluated in real-world package delivery ETA prediction and achieved a better performance than various adapted baselines.

Acknowledgments

This research has been funded in part by the USC Integrated Media Systems Center (IMSC) and Alibaba Group through Alibaba Research Fellowship Program. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsor.

References

1. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence **35**(8), 1798–1828 (2013)
2. Cao, S., Lu, W., Xu, Q.: Deep neural networks for learning graph representations. In: Thirtieth AAAI conference on artificial intelligence (2016)
3. Chang, B., Park, Y., Park, D., Kim, S., Kang, J.: Content-aware hierarchical point-of-interest embedding model for successive poi recommendation. In: IJCAI. pp. 3301–3307 (2018)
4. Chang, S., Han, W., Tang, J., Qi, G.J., Aggarwal, C.C., Huang, T.S.: Heterogeneous network embedding via deep architectures. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 119–128 (2015)
5. Dhillon, P., Foster, D.P., Ungar, L.H.: Multi-view learning of word embeddings via cca. In: Advances in neural information processing systems. pp. 199–207 (2011)
6. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 855–864 (2016)
7. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Advances in neural information processing systems. pp. 1024–1034 (2017)
8. Hamilton, W.L., Leskovec, J., Jurafsky, D.: Diachronic word embeddings reveal statistical laws of semantic change. arXiv preprint arXiv:1605.09096 (2016)
9. Hinton, G.E., McClelland, J.L., Rumelhart, D.E., et al.: Distributed representations. Carnegie-Mellon University Pittsburgh, PA (1984)
10. Ke, J., Yang, H., Zheng, H., Chen, X., Jia, Y., Gong, P., Ye, J.: Hexagon-based convolutional neural network for supply-demand forecasting of ride-sourcing services. IEEE Transactions on Intelligent Transportation Systems (2018)
11. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in neural information processing systems. pp. 2177–2185 (2014)
12. Levy, O., Goldberg, Y., Dagan, I.: Improving distributional similarity with lessons learned from word embeddings. Transactions of the Association for Computational Linguistics **3**, 211–225 (2015)

13. Li, Y., Fu, K., Wang, Z., Shahabi, C., Ye, J., Liu, Y.: Multi-task representation learning for travel time estimation. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. pp. 1695–1704 (2018)
14. Li, Y., Yang, M., Zhang, Z.: A survey of multi-view representation learning. IEEE transactions on knowledge and data engineering **31**(10), 1863–1883 (2018)
15. Liu, X., Liu, Y., Li, X.: Exploring the context of locations for personalized location recommendations. In: IJCAI. pp. 1188–1194 (2016)
16. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
18. Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., Ng, A.Y.: Multimodal deep learning. In: ICML. pp. 689–696 (2011)
19. Niemeyer, G.: Geohash (2008)
20. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 701–710 (2014)
21. Sahlgren, M.: The distributional hypothesis. Italian Journal of Disability Studies **20**, 33–53 (2008)
22. Shuman, D.I., Narang, S.K., Frossard, P., Ortega, A., Vandergheynst, P.: The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE signal processing magazine **30**(3), 83–98 (2013)
23. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 945–953 (2015)
24. Tang, J., Qu, M., Mei, Q.: Pte: Predictive text embedding through large-scale heterogeneous text networks. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 1165–1174 (2015)
25. Teng, S.H.: Scalable algorithms for data and network analysis. Foundations and Trends in Theoretical Computer Science **12**(1–2), 1–274 (2016)
26. Tobler, W.R.: A computer movie simulating urban growth in the detroit region. Economic geography (1970)
27. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 1225–1234 (2016)
28. Wang, W., Arora, R., Livescu, K., Bilmes, J.: On deep multi-view representation learning. In: International Conference on Machine Learning. pp. 1083–1092 (2015)
29. Wu, F., Wu, L.: Deepeta: A spatial-temporal sequential neural network model for estimating time of arrival in package delivery system. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 774–781 (2019)
30. Xia, F., Liu, T.Y., Wang, J., Zhang, W., Li, H.: Listwise approach to learning to rank: theory and algorithm. In: Proceedings of the 25th international conference on Machine learning. pp. 1192–1199. ACM (2008)
31. Yao, D., Zhang, C., Zhu, Z., Huang, J., Bi, J.: Trajectory clustering via deep representation learning. In: 2017 international joint conference on neural networks (IJCNN). pp. 3880–3887. IEEE (2017)
32. Zhang, J.D., Chow, C.Y., Li, Y.: igeorec: A personalized and efficient geographical location recommendation framework. IEEE Transactions on Services Computing **8**(5), 701–714 (2014)